First Contact: an Active Vision Approach to Segmentation

P. Fitzpatrick
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
paulfitz@ai.mit.edu

Abstract— **How a robot should grasp an object depends on its size and shape. Such parameters can be estimated visually, but this is fallible, particularly for unrecognized, unfamiliar objects. Failure will result in a clumsy grasp or glancing blow against the object. If the robot does not learn something from the encounter, then it will be apt to repeat the same mistake again and again. This paper shows how to recover information about an object's extent by poking it, either accidentally or deliberately. Poking an object makes it move, and motion is a powerful cue for visual segmentation. The periods immediately before and after the moment of impact turn out to be particularly informative, and give visual evidence for the boundary of the object that is well suited to segmentation using graph cuts. The segmentation algorithm is shown to produce results consistent enough to support autonomous collection of datasets for object recognition, which enables often-encountered objects to be segmented without the need for further poking.**

I. INTRODUCTION

Object recognition and image segmentation are intertwined problems, since each is far easier to do if we can perform the other. Yet it is important to be able to segment unfamiliar objects – for example, to guide a manipulator to grasp them. Rather than simply failing in visually ambiguous situations, an active robotic platform has the potential to perform experiments on its environment that resolve the ambiguity (see Figure 1). Methods for characterizing the shape of an object through tactile information have been developed, such as shape from probing [3], [8] or pushing [6], [7]. In this paper, we show that the *visual* feedback generated when the robot moves an object is highly informative, even when the motion is short and poorly controlled, or even accidental. This opens the door to extracting information from failed actions such as a glancing blow to an object during an attempt at manipulation, potentially giving the robot the data it needs to do better next time. We adopt the term "poking" (as opposed to "probing") for such actions, to convey the idea of a quick jab to evoke visual data instead of an extended grope for tactile data. Although tactile and visual information could usefully be combined, no tactile or proprioceptive information is assumed in this paper – not even to determine whether the robot is in contact with an object.

But how useful is a segmentation method that relies on action, in practice? It would be cumbersome to always have to poke around to segment an object each time it comes into view. But the cleanly segmented views of objects generated by poking are exactly what is needed to train up an object recognition system, which in turn makes segmentation *without* further poking possible. So the kind of active segmentation proposed here can serve as an online teacher for passive segmentation techniques. Analogously, while an experienced adult can interpret visual scenes perfectly well without acting upon them, linking action and perception seems crucial to the developmental process that leads to that competence [4].

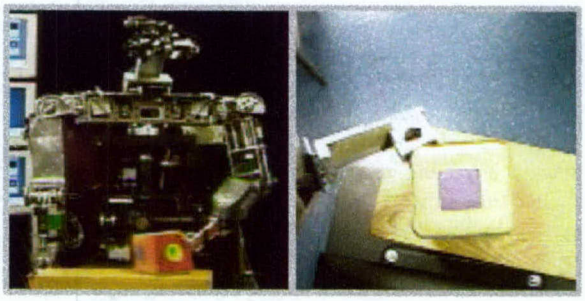


Fig. 1. A motivating scenario. The robot (left) reaches towards an object in its environment while fixating it with a camera. The robot's view is shown on the right. The boundary between the cube and the table it is sitting on is clear to human eyes, but too subtle to be reliably segmented by current automatic methods. But once the robot arm comes in contact with the object, it can be easily segmented from the background using the motion due to the impact.

II. FIRST CONTACT

If the object is to be segmented based on motion, we need to differentiate its motion from any other sources in the scene – particularly that of the robot itself. A high-quality opportunity to do this arises right at the moment of first contact between the robot and the object. This contact could be detected from tactile information, but it is also straightforward to detect visually, which is the method described here. The advantage of using visual information is that the same techniques can be applied to contact events about which the robot has no privileged knowledge, such as a human hand poking an object (see Section VI).

For real-time operation, the moment of contact is first detected using low-resolution processing, and then the

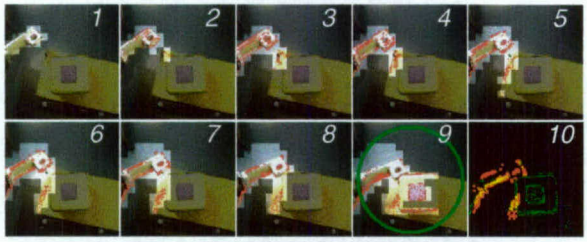


Fig. 2. The moment of (ground) truth – detecting the point of impact between the robot's arm and an object. As the arm swings in, its motion is tracked frame by frame and aggregated within relatively low-resolution bins (highlighted squares). When an implausibly large spread in motion is detected across these bins, higher resolution processing is activated and segmentation begins.

images before and after the contact are subjected to more detailed (and slower) analysis as described in the following section. Figure 2 shows a visualization of the procedure used. When the robot is attempting to poke a target, it suppresses camera movement and keeps the target fixated for maximum sensitivity to motion. A simple Gaussian model is maintained for the (R,G,B) color values of each pixel, based on their value over the last ten frames (one third of a second) received. Significant changes in pixel values from frame to frame are detected and flagged as possible motion. As the arm moves in the scene, its motion is tracked and discounted, along with its shadow and any background motion. Any area that the arm moves through is marked as "clear" of the object for a brief period – but not permanently since the arm may cross over the object before swinging back to strike it. An impact event is detected through a signature explosion of movement that is connected with the arm but spread across a much wider distance than the arm could reasonably have moved in the time available. Since the object is stationary before the robot pokes it, we can expect the variance of the Gaussians associated with the individual pixel models to be low. Hence they will be very sensitive to the pixel value changes associated with the sudden motion of the object. Once the impact is detected, we can drop briefly out of real-time operation for a few seconds and perform the detailed analysis required to actually cleanly segment the object based on the apparent motion.

III. Figure/Ground Separation

Once the moment of contact is known, the motion visible before contact can be compared with the motion visible after contact to isolate the motion due to the object. Since we observe pixel variation rather than true motion, we can also factor in how we expect them to relate – for example, a highly textured region with no observed change over time can be confidently declared to be stationary, while a homogeneous region may well be in motion even if there is little observed change. In general, the information we have is sparse in the image and can be framed as probabilities that a pixel belongs to the foreground (the object) or the background (everything else). Let us first look at a simpler version of this problem, where for those pixels that we do have foreground/background information, we are completely confident in our assignments.

Suppose we have some information about which pixels in an image $I(x,y)$ are part of the foreground and which are part of the background. We can represent this as:

$$A(x,y) = \begin{cases} -1, & I(x,y) \text{ is background} \\ 0, & I(x,y) \text{ is unassigned} \\ 1, & I(x,y) \text{ is foreground} \end{cases}$$

We now wish to assign *every* pixel in the image to foreground or background as best we can with the sparse evidence we have. One approach would be to create a cost function to evaluate potential segmentations, and choose the segmentation with minimum cost. If we are willing to accept constraints on the kind of cost function we can use, then there is a family of maximum-flow/minimum-cut algorithms that can provide good approximate solutions to this problem [2]. To apply them, we need to translate our problem into the form of a graph, as shown in Figure 3. Each pixel maps to a node in the graph, and is connected by edges to the nodes that represent neighboring pixels. There are two special nodes corresponding to the labels we wish to assign to each pixel (foreground or background). The problem the minimum-cut algorithms can solve is how to split this graph into two disjoint parts, with the foreground node in one and the background node in the other, such that the total cost of the edges broken to achieve this split is minimized. So our goal should be to assign costs to edges such that a minimum cut of the graph will correspond to a sensible segmentation.

Let $N(x,y)$ be the node corresponding to pixel $I(x,y)$. Let N_{+1} be the node representing the foreground, and N_{-1} be the node representing the background. If we are completely confident in our classification of pixel $I(x,y)$ into background or foreground, we may encode this knowledge by assigning infinite cost to the edge from $N(x,y)$ to $N_{A(x,y)}$ and zero cost to the edge from $N(x,y)$ to $N_{-A(x,y)}$.

$$\mathscr{C}(N(x,y),N_{+1}) = \begin{cases} \infty, & A(x,y) = 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\mathscr{C}(N(x,y),N_{-1}) = \begin{cases} \infty, & A(x,y) = -1 \\ 0, & \text{otherwise} \end{cases}$$

This will force the minimum-cut algorithm to assign that pixel to the desired layer. In practice, the visual information will be more ambiguous, and these weights should be correspondingly "softer".

Costs also need to be assigned to edges between pixel nodes. Suppose we expect foreground information to be available most reliably around the edges of the object,

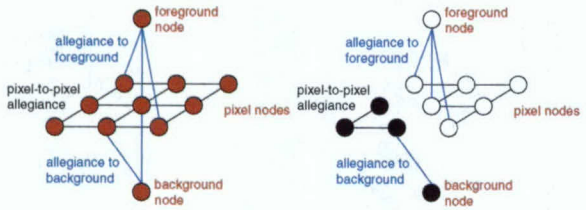


Fig. 3. For a two-label problem on a 2D image, the input to a minimum-cut algorithm is typically as shown on the left. There is a node for each pixel, and two special nodes corresponding to the labels (foreground/background). Visual evidence is encoded on edges between the nodes. The output of the algorithm is shown on the right. The graph is cut into two disjoint sets, each containing exactly one of the special nodes, such that the total cost of the edges cut is (approximately) minimized.

as is in fact the case for motion data. Then a reasonable goal would be to use the minimum cut to minimize the total perimeter length of segmented regions, and so merge partial boundary segments into their bounding region. To do this, we could simply assign the actual 2D Euclidean distance between the pixels as the cost. This is not quite sufficient if our edge information is noisy, because it permits almost "zero-area" cuts around individual isolated foreground pixels. We need to place an extra cost on cutting around a foreground pixel so that it becomes preferable to group near-neighbors and start generating regions of non-zero area. For this example, we simply double the cost of cutting edges that are connected to pixels known to be foreground or background.

$$\mathscr{C}(N(x_0,y_0),N(x_1,y_1)) = \begin{cases} D, & A(x_0,y_0)=0, \\ D, & A(x_1,y_1)=0 \\ 2D, & \textit{otherwise} \end{cases}$$

$$\text{where } D = \sqrt{(x_0-x_1)^2+(y_0-y_1)^2}$$

Edges are only placed between neighboring pixel nodes, to prevent an explosion in connectivity. A neighborhood is defined as shown in Figure 4.

Figure 5 shows examples of minimum-cuts in operation. The first image (top left) has two (noisy) lines of known foreground pixels, of length w. The minimum cut must place these pixels inside a foreground region. If the regions are disjoint, the total perimeter will be at least $4w$. If the the lines are instead placed inside the same region, the cost could be as little as $2w+2h$ where h is the distance between the two lines, which is less than w. The figure shows that this is in fact the solution the minimum-cut algorithm finds. The next two examples show what this minimum perimeter criterion will group and what it will leave separate. The fourth example shows that by introducing known background pixels, the segmentation can change radically. The patch of background increases the perimeter cost of the previous segmentation by poking a hole in it that is large enough to tip the balance in favor of individual rather than merged regions. This basic formulation can be extended without difficulty to natural data, where foreground/background assignments are soft.

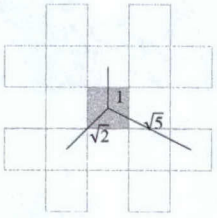


Fig. 4. The segmentation algorithm is sensitive to the length of the perimeters around foreground regions. It is important that the local pixel connectivity not be so sparse as to introduces artifacts into that perimeter. For example, suppose we just used 4-connected regions. The cost of a zig-zag approximation to a diagonal edge would be $\sqrt{2}=1.41$ times what it ought to be. 8-connected regions are better, but still distort the perimeter cost significantly, up to a factor of $\frac{1+\sqrt{2}}{\sqrt{5}}=1.08$. The neighborhood shown here, which is 8-connected plus "knight moves", introduces a distortion of at most $\frac{1+\sqrt{5}}{\sqrt{10}}=1.02$. Further increases in neighborhood size increases computation time without bringing significant benefit.

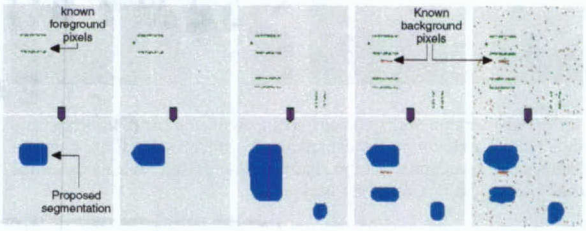
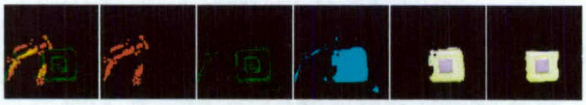


Fig. 5. Some simple segmentation examples. Input images are shown on the upper row, output is shown as filled regions on the lower row. In the first three cases, the border of the image is set to be background, and the dark pixels are foreground. In the fourth case, a small extra patch of pixels known to be in the background is added, which splits the large segmented region from the previous case in two. The final case shows that the algorithm is robust to noise, where 1% of the pixels are assigned to foreground or background at random. This is in fact a very harsh kind of noise, since we have assumed complete certainty in the data.

IV. Before and After

The previous section showed that if there is some evidence available about which pixels are part of the foreground and which are part of the background, it is straightforward to induce a plausible segmentation across the entire image. Figure 6 shows an example of how the necessary visual evidence is derived in practice. The statistical significance of changes in pixel values (the "apparent motion") is measured in the frames directly following the contact event, using the continuously updated Gaussian models. The measurements are combined over two frames to avoid situations where the contact event occurs just before the first frame, early enough to generate enough motion for the contact event to be detected but late enough not to generate enough motion for a successful



Motion in frame immediately after impact | *Aligned motion from before impact* | *Masking out prior motion* | *Result of segmentation* | *Largest connected region found* | *Refinement of segmentation*

Fig. 6. Collecting the motion evidence required for segmentation. The apparent motion after contact, when masked by the motion before contact, identifies seed foreground (object) regions. Such motion will generally contain fragments of the arm and environmental motion that escaped masking. Motion present before contact is used to identify background (non-object) regions. This prevents the region assigned to the object motion from growing to include these fragments. The largest connected region, with a minor post-processing clean-up, is taken as the official segmentation of the object.

Fig. 7. Challenging segmentations. The example on the right, a blue and white box on a glossy poster, is particularly difficult since it has complex shadows and reflections, but the algorithm successfully distinguishes the white part of the box from the background.

segmentation. The frames are aligned by searching for the translation that best matches the apparent motion in the two frames (rotation can be neglected for these very short intervals). A similar measurement of apparent motion from immediately before the contact event is also aligned, and is used to partially mask out motion belonging to the robot arm, its shadow, and unrelated movement in the environment. The remaining motion is passed to the segmentation algorithm by giving pixels a strong "foreground" allegiance (high cost on edge to special foreground node). Importantly, the motion mask from before contact is also passed to the algorithm as a strong "background" allegiance (high cost on edge to background node). This prevents the segmented region from growing to include the arm without requiring the masking procedure to be precise. The maximum-flow implementation used is due to [2].

Perimeter-minimization seems particularly appropriate for the kind of motion data available, since for textureless objects against a textureless background (the worst case for motion segmentation) motion is only evident around the edges of the object, with a magnitude that increases with the angle that edge makes to the direction of motion. A textured, cluttered background could only make life simpler, since it makes it easier to confidently assert that background regions are in fact not moving.

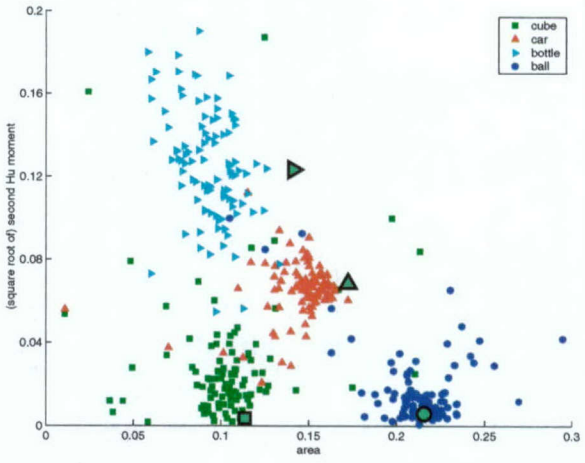


Fig. 8. A large collection of segmentations are clustered by color histogram, assigned human-readable labels based on the object that occurs most frequently in each cluster, and then plotted (area versus second Hu moment). The enlarged markers show hand-segmented reference values. The segmentations are quite consistent, although area tends to be a fraction smaller than in the hand-segmented instances.

V. Experimental results

How well does active segmentation work? The segmentation in Figure 6 is of the object shown in the introduction (Figure 1), a cube with a yellow exterior sitting on a yellow table. Active segmentation has a clear advantage in situations like this where the color and texture difference between object and background would be too small for conventional segmentation but is sufficient to generate apparent motion when the object is poked (see Figure 7 for more examples).

Active segmentation was recruited as a developmentally plausible means of initiating early integration of vision and manipulation as part of a large-scale experiment aimed at implementing a robotic analogue of the mirror-neuron system found in primates [4]. The robot was given a poking behavior so that it would extend its arm to swing near anything reachable that its attention system was directed towards. A human caregiver brought interesting objects to the robot to poke. The objects differed in how they rolled, and the robot learned to exploit that fact. Active segmentation played two roles in this experiment: collecting data for later object recognition and localization, and providing a good segmentation for tracking the motion of the object after contact. End to end performance of the system was described in [4]. Here we report on the performance of the active segmentation component in isolation.

By clustering segmented views of objects based on color histograms, the robot collected about 100 views of each object. Since the segmented shape was not used in

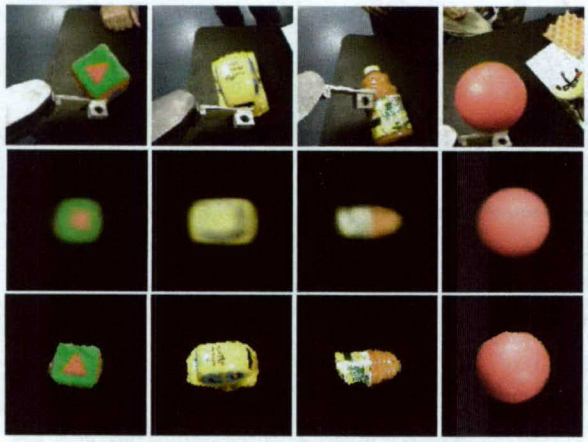


Fig. 9. The top row shows the four objects used in this experiment, seen from the robot's perspective. The middle row shows prototypes derived for those objects using a naïve alignment procedure. None of the prototypes contain any part of the robot's manipulator, or the environment. These prototypes are used to find the best available segmentations of the objects (bottom row).

clustering, it can be used as an independent measure of cluster quality. A simple way to characterize shape is with area and the Hu moments [5], which are invariant to translation and in-plane rotation. Figure 8 shows the area plotted against the second Hu moment (a measure of anisotropy) for all the segmentations recorded. The second Hu moment Φ_2 for a region R with centroid (x_0, y_0) and area μ_{00} is:

$$\begin{aligned} \Phi_2 &= (\nu_{20} - \nu_{02})^2 + 4\nu_{11}^2 \\ \nu_{pq} &= \frac{1}{\mu_{00}^2} \int\int_R (x - x_0)^p (y - y_0)^q dx dy \end{aligned}$$

Leave-one-out cross validation on a simple nearest neighbor classifier gives a classification accuracy of 90.8%. So the shape information is a good predictor of the color histogram labelling, presumably because they are both reliable functions of object identity. If the quality of the clusters generated is sufficiently good, it should be possible to extract reliable consensus prototypes for each object. This is in fact the case, as Figure 9 shows. Using the most naive alignment procedure and averaging process possible, a blurry "mean" view of the objects can quickly be derived. This could be sharpened by better alignment procedures, or just used to pick out the best single match to the mean view for each object. Of course, this paper is not proposing that Hu moments and simple color histograms are how recognition should be done – there are better ways (for example, see [9]), rather it is giving evidence that active segmentation can generate data of sufficient quality to train up a recognizer.

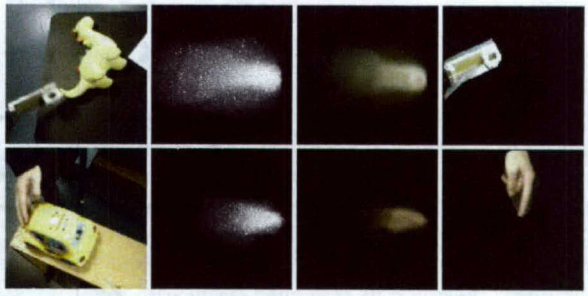


Fig. 10. The robot manipulator (top left) was automatically segmented during 20 poking sequences. The segmentations were aligned and averaged, giving the mask and appearance shown in the adjacent images. The best matching view is shown on the top right. A similar result for the human hand is shown on the bottom, based on much less data (5 poking sequences, hands of two individuals).

VI. Operational vision

In a sense, poking provides the robot with an operational definition of what objects are by giving it an effective procedure for learning about them. It is not perfect – for example, the robot is effectively blind to objects that are too small or too large – but for objects at an appropriate scale for manipulation, it works well. Once the robot is familiar with a set of such objects, we can go further and provide an operational definition of a *manipulator* as something that acts upon these objects. We can create an effective procedure for learning about manipulators by simply giving the robot a predisposition to fixate familiar objects. This enables the same machinery developed for active segmentation to operate when a foreign manipulator (such as the human hand) pokes the fixated object. Of course the robot can easily distinguish segmentations of its own arm from that of others simply by checking whether it was commanding its arm to move towards the target at the time. The manipulator can be segmented by hypothesizing that it moves towards the object at a constant velocity in the period immediately preceding the moment of contact. Estimating the velocity from the gross apparent motion allows the segmentation problem to be expressed in the form introduced in Section III, where the foreground is now taken to be regions moving at the desired velocity, and the background is everything else. Figure 10 shows preliminary results for this procedure. The results are based on relatively little data, yet are already sufficient to pick out good prototype views for the robot and human manipulator. A procedure like this could be used to autonomously train a recognizer for the human hand, which could then be included in further operational definitions, expanding the robot's domain of grounded knowledge ever outwards – but this is very much future work.

VII. Summary and Conclusions

While it has long been known that motor strategies can aid vision [1], work on active vision has focused almost exclusively on moving cameras. There is much to be gained by recruiting a manipulator to aid perception. This paper has shown that unfamiliar objects can be reliably segmented by simply tapping them and watching how they move. Many extensions to this strategy are possible. For example, a robot might try to move an arm around *behind* the object, to reveal its occluding boundary.

Active segmentation gives clear results for a rigid object that is free to move. What happens for non-rigid objects and objects that are attached to other objects? Here the results of poking are likely to be more complicated to interpret – but in a sense this is a good sign, since it is in just such cases that the idea of an object becomes less well-defined. Poking has the potential to offer an operational theory of "object-hood" that is more tractable than a vision-only approach might give, and which cleaves better to the true nature of physical assemblages.

VIII. Acknowledgements

Funds for this project were provided by DARPA under contract number DABT 63-00-C-10102, and by the Nippon Telegraph and Telephone Corporation.

IX. References

[1] J.Y. Aloimonos, I. Weiss, and A. Bandopadhay. Active vision. *International Journal on Computer Vision*, 2:333–356, 1987.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 359–374, 2001.

[3] R. Cole and C. Yap. Shape from probing. *Journal of Algorithms*, 8(1):19–38, 1987.

[4] P. Fitpatrick and G. Metta. Towards manipulation-driven vision. In *IEEE/RSJ Conference on Intelligent Robots and Systems*, 2002.

[5] M. K. Hu. Visual pattern recognition by moment invariants. In *IRE Transactions on Information Theory 8*, pages 179–187, 1962.

[6] Yan-Bin Jia and Michael Erdmann. Observing pose and motion through contact. In *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1998.

[7] M. Moll and M. A. Erdmann. Reconstructing shape from motion using tactile sensors. In *Proc. 2001 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Maui, HI, October/November 2001.

[8] E. Paulos. Fast construction of near optimal probing strategies. Master's thesis, University of California, Berkeley, 1999.

[9] B. Schiele and J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, January 2000.